

Описание технической архитектуры  
программного обеспечения  
«СмартРезерв. Автоматизация учета  
инфраструктуры»

ООО «РУКСофт»  
г. Воронеж  
2024 г.

Установка программного обеспечения «СмартРезерв. Автоматизация учета инфраструктуры» включает в себя следующие пункты:

1. [Установка Docker CE на Ubuntu Server 18.04](#)
2. [Запуск контейнеров через docker compose](#)

## Установка Docker на Ubuntu Server 24.04

1. Настройка Docker apt репозитория:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
echo \
"deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

2. Установка пакетов docker с версией 5:26.1.0-1~ubuntu.24.04~noble:

```
VERSION_STRING=5:26.1.0-1~ubuntu.24.04~noble

sudo apt-get install docker-ce=$VERSION_STRING docker-ce-
cli=$VERSION_STRING containerd.io docker-buildx-plugin docker-
compose-plugin
```

3. Проверяем, что установка docker и docker compose прошла успешно:

```
docker version
docker compose version
```

При успешной установке в консоли появится текст с версиями компонентов.

## Запуск контейнеров с помощью docker compose

1. Создаем директорию для работы с приложением:

```
mkdir aim-remsoft
cd aim-remsoft
```

2. В директории aim-remsoft создаем файл docker-compose.yml (vi docker-compose.yml) и добавляем в него описание контейнеров:

```
version: '2.4'
```

```
services:
```

```
  postgres:
```

```
    container_name: "postgres"
    image: postgres:11.21-alpine3.18
    mem_limit: 512m
    mem_reservation: 256M
    expose:
      - 5432
    ports:
      - 5432:5432
    volumes:
      - postgres-data:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: ${POSTGRES_DB}
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres -h 127.0.0.1"]
      interval: 10s
      timeout: 10s
      retries: 6
      start_period: 30s
    restart: unless-stopped
```

```
  db-migration:
```

```
    container_name: "db-migration"
    image: migration:dev-latest
    environment:
      DB_URL: ${POSTGRES_URL}
      DB_USERNAME: ${POSTGRES_USER}
      DB_PASSWORD: ${POSTGRES_PASSWORD}
    depends_on:
      postgres:
        condition: service_healthy
    restart: "no"
```

```
  backend:
```

```
    container_name: "backend"
    image: backend:aim-rmcsoft
    ports:
      - 8080:8080
    environment:
      DB_URL: ${POSTGRES_URL}
```

```

    DB_USERNAME: ${POSTGRES_USER}
    DB_PASSWORD: ${POSTGRES_PASSWORD}
    LDAP_ENABLED: "false"
    DOMAIN: "localhost"
    CLIENT_URL: "http://localhost:80"
    APP_ACTIVE_PROFILE: "no-kafka, no-kafka-producer"
    GRAVATAR_PATH: ""
  depends_on:
    postgres:
      condition: service_healthy
    db-migration:
      condition: service_completed_successfully

  frontend:
    container_name: "frontend"
    image: frontend:aim-rmcsoft
    ports:
      - 80:80
    environment:
      SSL_ENABLED: "false"
      BACKEND_ADDR: "backend:8080"
      SERVER_NAME: "localhost"
    depends_on:
      backend:
        condition: service_started

  volumes:
    postgres-data:
      driver: local

  networks:
    default:
      driver: bridge

```

3. В директории aim-remsoft создаем файл .env (vi .env) для описания переменных окружения и добавляем в него следующие переменные:

```

POSTGRES_USER=postgres
POSTGRES_PASSWORD=password
POSTGRES_DB=aim
POSTGRES_URL=postgres:5432

```

4. Для запуска приложения, находясь в директории aim-remsoft, в консоли необходимо выполнить команду:

```
docker compose up
```

5. Если приложение успешно запустилось, в консоли появятся логи с состоянием сервисов. Приложение будет доступно по адресу <http://localhost:80>.